Exhibit 1

Preliminary UMP Viewer Application Functional Specification & Software Requirements

UNIVERSAL

Executive Summary

This document describes the specification and functionality of the UMG player. The document has three sections each addressing a part of the player and its operation. This document does not attempt to provide a fully comprehensive detailed specification of the player or its operation, but rather to provide a framework in which the player can be developed.

The three sections of the document are:

- Section One   Architecture
- Section Two   Functions
- ·Section Three   Environment

It is anticipated this document will be used to refine the design and will be iterated in line with the development of the player. It is intended that this development itself be iterative in nature and will have the following phases:

Phase One	Non functional GUI prototypes
Phase Two	Non functional prototype including iterated GUI components
Phase Three	Partially functional prototype with feature complete GUI and specification
Phase Four	Feature complete player in Alpha state
Phase Five	Feature complete pre production player in Beta State
Phase Six	Feature complete production player for General Release

This document is designed to evolve with these phases with the immediate outcome of the completion of this document being the resourcing of the development team to achieve Phases One to Six.

Revision and Iteration History

Version	Publication date	Authors	Summary of Changes and updates

REDACTED

Table of Contents:

Introduction

This document describes the Universal Player and those components required to integrate with the █ █████████████ components. The player has been broken into a number of segments, which are outlined below:

Each of segments, the components within them and the functions they undertake will be specified in the final design document. This document describes them within the constraints of the information available at this time.

A significant part of the design process will involve the development of some key components which will be prototyped as proof of concept.

The document has two sections, the description of the components involved in the player and the functional description of the actions and interactions of those components.

Design Goals

The objective of the player will be to provide an intuitive & easy to use interface stimulating the consumption of music in new and innovative ways. This will be achieved by consumers interacting with an electronic music commerce service in which the content and its associated business rules are delivered in secure (DigiBox) containers.
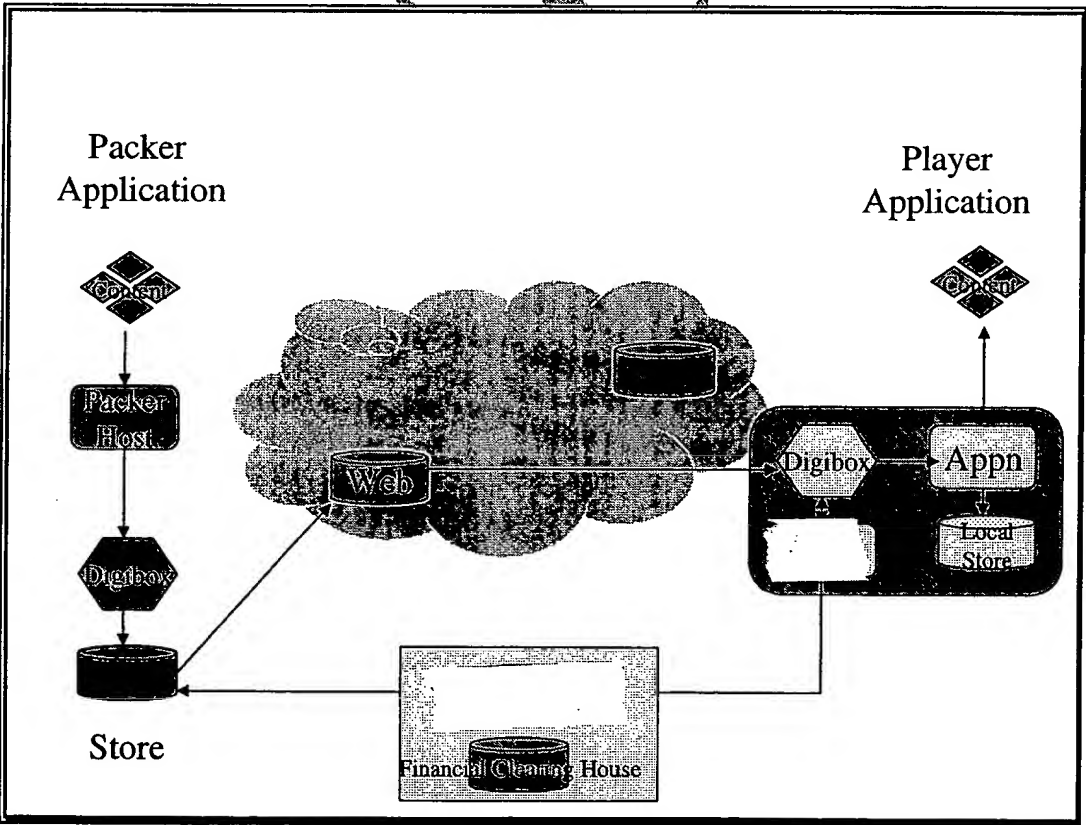
The player will use an object oriented model to describe and implement its functionality with the goals of achieving robust design & code, maximizing implementation speed , developing reusable code & minimizing the size of the code as much as possible.

The design goals have both strategic and tactical objectives:

Strategic:

The goal of this project is to produce a player that can be used for the consumption of music delivered by an e-commerce (EMD, Electronic Music Distribution) system. Within this the technical specification of the application will be undertaken within these principles and guidelines:

1. The player will attempt to remain as device independent in concept as possible.

2. All code will be written with a view to establishing a set of reusable code libraries which will facilitate the process of porting this functionality to as many platforms as possible, including more simplistic Operating Systems such as those used in Consumer Electronic (CE) devices.

3. The code will be written in Visual C++ for maximum compatibility. Thought and consideration was put into developing the UMP viewer with the Java language. At this time due to the tardy execution speeds of Java versus C++ it was decided not to pursue an initial Java implementation.

4. The design will at all times attempt to successfully leverage existing code sources , except where this will compromise the integrity of the product or too closely bind the implementation to the Windows system.

5. All calls to the ███████ systems will be via an interface abstraction to minimize and contain dependence on these technologies.

Tactical:

The main tactical design goals are to ensure the user interface is intuitive and not constrained within the Windows look and feel, subject to this being the initial host platform. Some goals are:

1. Eliminate scrolling windows as much as possible by using contextual sensitivity to size the window appropriately whenever feasible.

2. Again using contextual sensitivity, minimize the number of keystrokes and mouse movements as much as possible.

3. Combine simplistic UI choices for most commonly used functions with more detail and greater flexibility one click under the surface. For example, clicking the "Web" button on an audio screen takes you to the default Web page but "Open" in the same context spawns a window with links to all the associated sites.

Section one: Architecture

REDACTED

This page is left Intentionally Blank

Player Architecture

The Universal Media Player will be a self contained system providing the facilities for playing music and associated content & data delivered and paid for by the ███████████████. The application will be dependent on the functionality offered by the █████████████████████████████ software system. The overview of these is described in the Technical Overview Document (TOD).

The components of the systems are expressed in terms of their Interface, "███████████████████ and preferred embodiment.

The detailed functionality of the application cannot in most cases yet be defined as the information available to us at this point in time and maturity of █████████ tool sets is not sufficient to describe theses detailed interactions. Our access to detailed technical information about ████████ has been constrained because of continuing contract negotiations. This document presents the view as far as possible at present of the player functionality and will be iterated in line with the availability of information from ████████

Diagram One-Player Overview
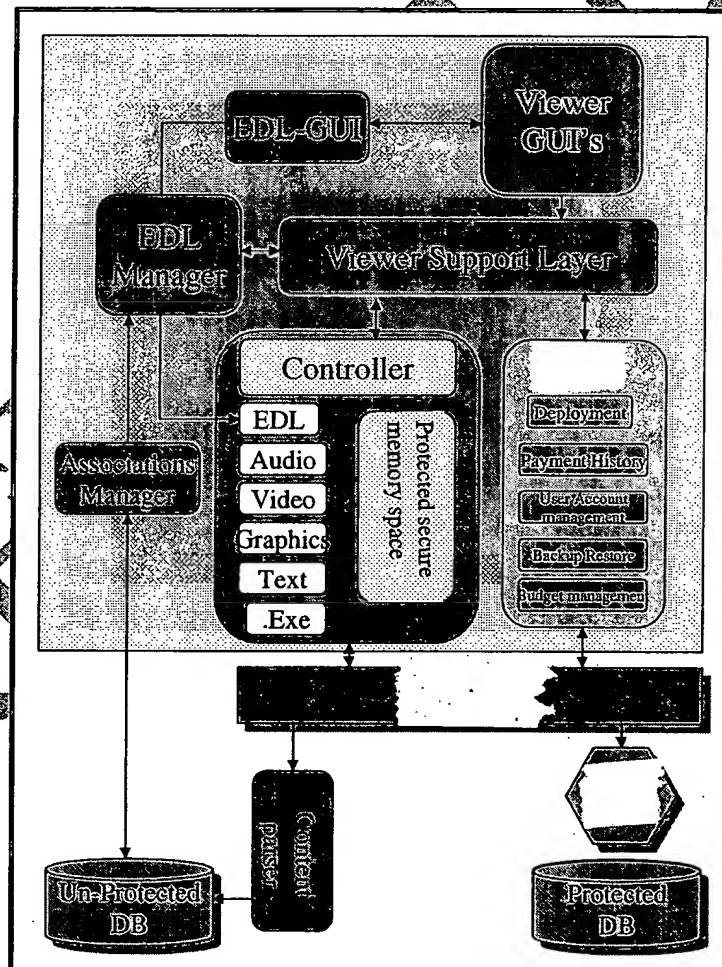
The Universal Media Player (UMP) will be able to view, play or execute multiple types of content. Supported types will be audio data in PAC/ACC and DTS formats, video in AVI, QuickTime & MPEG1 formats, JPEG,

BMP & GIF pictures, text in HTML, RTF, and ASCII formats and Windows 95 compatible executable & object files. UMP will also be able to read and generate Edit Decision Lists (EDLs). EDLs will allow authors and users to describe simultaneous or sequential played sequences of the supported types of content and, the UMP will be able to play & record these sequences.

The UMP will also be able to initiate, respond to and synchronize Online activities. Web links, email, Instant Messaging and other synchronization will be through the Edit Decision Lists (EDL's). The UMP will also be able to associate objects of various types (i.e. audio, pictures, links, EDLs, etc.) with each other either as defined by the authors at the time of creation as well as by users after distribution. Associations can also be made dynamically using search criteria. Associations to non-local (Web) objects will be indicated as such using intuitive visual cues.

Content will be protected and business rules enforced using ███████████████. The assumption is that ███████ approved individual decoders for the different media-formats will exist or will exist by the necessary time.[2]

The player has the following component groups:

Application Component	Description
GUI	Graphical User Interface
Viewer Support Layer	Interface layer to ████████████████████████

Controller	Control management for the player includes:
	
	– Secure memory management component
	– Media players and their integration to trustworthy state
	  ➤ Audio AC3/AC3 & DTS
	  ➤ Video AVI, QuickTime & MPEG 1
	  ➤ Pictures JPEG, BMP, GIF & MPEG stills
	  ➤ Text HTML and RTF
	  ➤ Executables EXE
	– Initiator component
	– Associations controller component
	Event scheduler component

EDL
(Edit Decision Lists)
	EDL Manager includes:
	– EDL Editor GUI Interface
	EDL Control logic including interface and management of:
	  ➤ Email Client
	  ➤ Chat Client
	  ➤ "Buddy List" Client
	  ➤ Web Browser
	– Controller Interface

Installer	Setup and initialization screens
	Integration with ██████ Node installation-initialization procedures Review of current hardware platform for installed components

---

[1] For certain types of data it may not be possible to fully-protect the entire content because it may not be possible to unpack it quickly enough. We will know more about this when ███████ performs some tests.
[2] If a certain decoder is not planned for delivery it is expected that ███████████ will supply source code for a similar type of decoder by the necessary time.

███

Graphical User Interface

Overview of GUI Approach

The GUI system is designed to provide an easy to use and intuitive interface to the users of the UMP. This section describes the architecture and scope of the envisaged GUI interface in terms of the specific screens and controls presented to the user through these screens.

Diagram Two- GUI overview
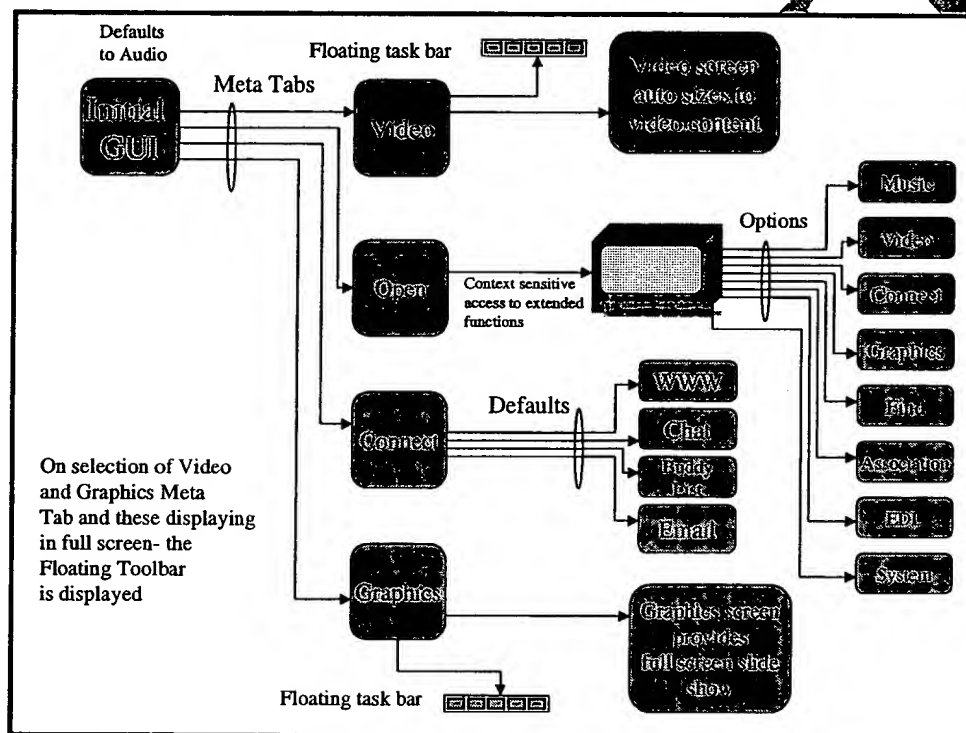
Navigation Principles

The initial screen can be launched in one of five modes, which are represented by the following icons:

-Music Icon   Connect Icon-

Video Icon-   Graphics Icon-

-Open Icon

Initially the default state will be the Audio state as shown below. When the screen is in a state, such as Audio, that icon will not be displayed, and only the other four options will be displayed.

Each screen will have context sensitive help available. This is not currently shown on the screen diagrams.

Audio GUI Screen.

The default initial screen will be the Audio screen. Thus provides the basic controls that the user will operate to control the player. The other possible meta selections are displayed across the top of the player. The picture shown in the player is the album artwork or other content creator selected artwork.

Diagram Three Audio GUI screen
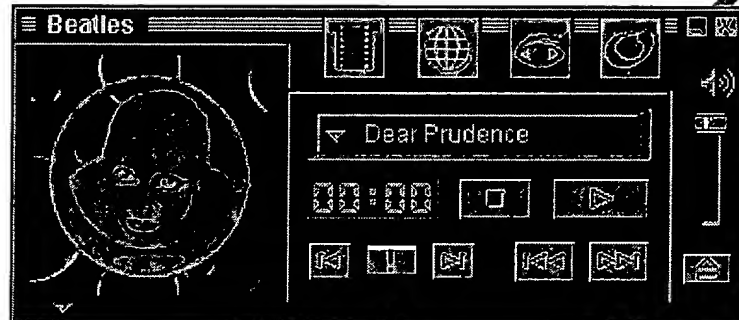
The primary operating controls are:

Audio Play/Pause Button- This is a rollover switch that selects either play or pause.

Stop Button- Selects the stop function

Next Song button-Selects the next song from the play list

Previous Song Button-Selects previous song from song list

Mark Location Button- Marks the current location for the EDL

Next Mark Button- Selects next mark

Previous mark Button Selects previous mark

Eject Button- terminates the current session

This is the volume control slider and mute button

The primary indicator controls are:
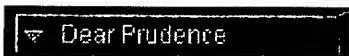
≡ Beatles ≡   Indicates the name of the play list

▽ Dear Prudence   This is a pop up window box containing the names of all the songs within
                             the play list. The currently playing song is displayed.

00:00   Time display panel

Video

The video screen will be a context sensitive display of the video material. This will involve the screen
automatically sizing to match the requirements of the content. An example of this is shown in the diagram
below.

Diagram Four   Video GUI screen

The primary operational controls are:

Video Slider with location indicator

Play /pause rollover button

Rewind Video button

Speaker Control button

The primary indicators are

≡ Beatles ≡   Indicates the name of the play list
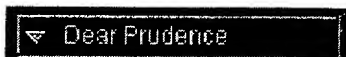
▽ Dear Prudence   This is a pop up window box containing the names of all the songs within
                                the play list. The currently playing song is displayed.

Connect

The connect screen provides the mechanism to make connections from the user to other users or sources of information. The connect meta tab is context sensitive and will add the three basic connect functions to the Audio or Video screen from which it is called. An example of the Audio screen with connect metatab selected is shown below.

Diagram Five   Connect GUI screen
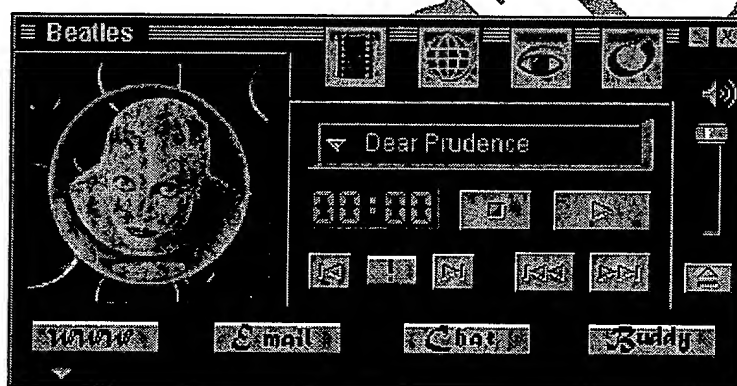
In this screen the four additional functions provided by the selection of the Connect Operation, which is still available as a selection are:

URL selector- Provides access to the web loaded with the default parameter(s)

E Mail selector- Provides access to e mail with the default locations loaded

Chat Selector- Provides access to the Chat functions with the default parameter(s)

Buddy List Selector- Provides access to the Buddy List functions

These buttons are designed to provide the user with quick and easy access to communications systems that add value to their music experience. Each function will have a default parameter set loaded from the Digibox which will be designed to give the user instant satisfaction. The current default settings are:

Operation	Parameter	Setting
WWW	URL	Home page of the band directed through the label server system
Email	email address	Mailbox for communications with the band hosted on the label server
Chat	IRC room selector	Home chat room for the band, user ID is their email address
Buddy	Instant message system identifier	Default buddy list maintained by the label on their servers

Each of these selections spawns the associated screen for the ▮▮▮ player.

There are a number of actions required to establish these connections that are handled on a first use basis. These are described in Section Two functional Requirements.

Alternate View States

In addition to the main screens described above, each screen may have two additional states,

- Minimised
- Extended

Within the current specification it is only intended for the Audio screen to have this functionality, although the full screen mode of both the Video and the Graphics screens may be considered their expanded states.

Diagram Six   Screen Views

	Audio	Video	Graphics
Expanded		Full Screen with floating tool bar	Full Screen with floating tool bar
Normal			
Minimized			
Task Bar			
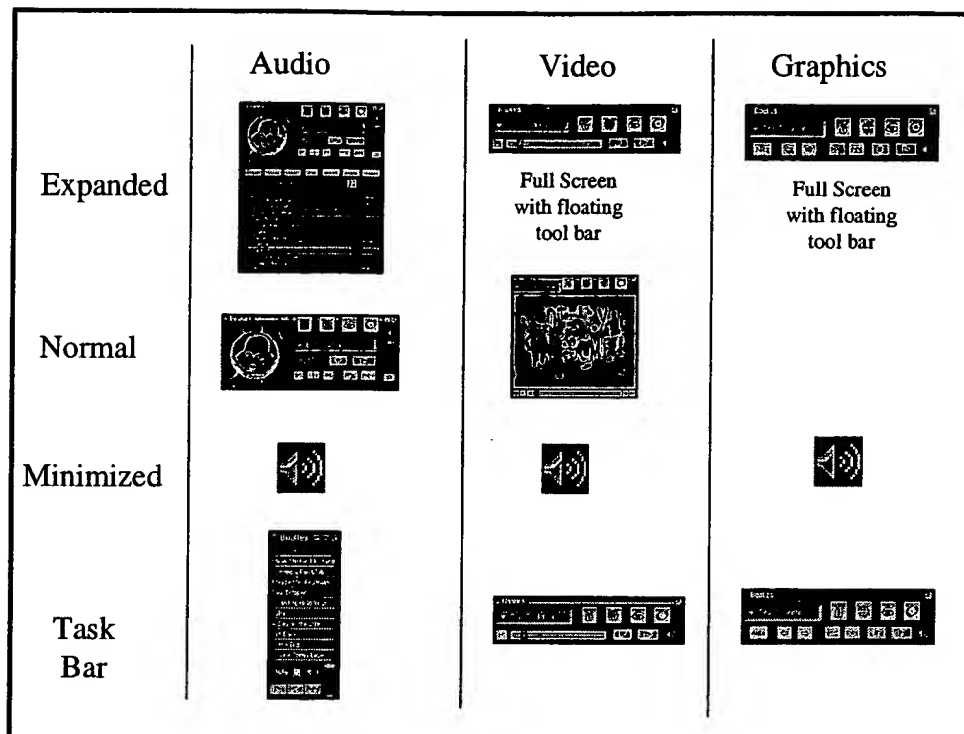
Task Bar views

There is a limited set of functionality available from the minimized view intended to give the user control when they are using other applications. Each of the three content display modes has a floating task bar which provides the relevant access. In the case of Video or Graphics these task bar views may be combined with full screen Graphics or Video images.

Audio Task Bar View

This task bar will be available form the speaker icon in the systems section of the task bar on the Windows hosts.

Diagram Seven Task bar view

Beatles

Dear Prudence
Mean Mister Mustard
Joining a Fan Club
Please Mr. Postman
Day Tripper
I want to Hold Your ...
Julia
A Day in the Life
Get Back
In the End
You're Gonna Lose ...

Mute
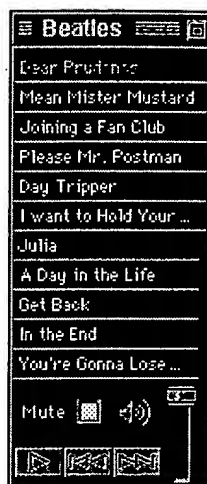
The operational controls provided are:

Audio Play/Pause Button- This is a rollover switch that selects either play or pause.

Next Song button-Selects the next song from the play list.

Previous Song Button-Selects previous song from song list.

Speaker Icon

Mute Button

There is also a volume slider.

Video Task bar View

If the full screen option is selected then the video controls will transform into a floating toolbar as shown in the diagram below. The Meta controls are also available as icons.

Diagram Eight   Video task bar view

Beatles

Dear Prudence
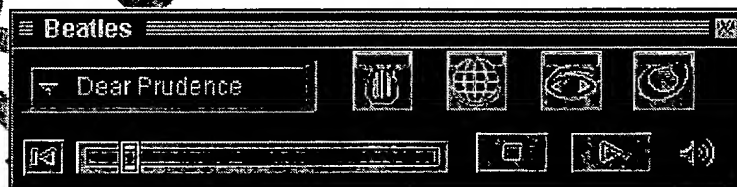
Within this screen the primary operational controls are:

Audio Play/Pause Button- This is a rollover switch that selects either play or pause.

Stop Button- Selects the stop function

Video Slider with location indicator

Rewind Video button

Speaker Control button

The primary indicators are

≡ Beatles ≡   Indicates the name of the play list
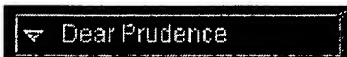
▽ Dear Prudence   This is a pop up window box containing the names of all the songs within the play list. The currently playing song is displayed.

Graphics Task bar View

Diagram Nine   Graphics Task bar View
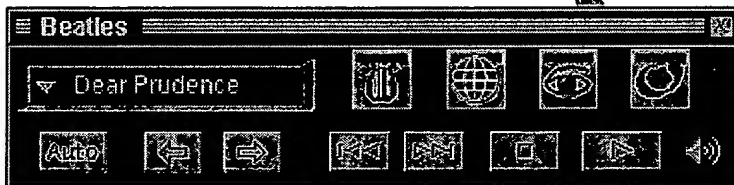
Within this screen the primary operational controls are:

▷ Graphics Play/Pause Button- This is a rollover switch that selects either play or pause.

□ Stop Button- Selects the stop function

Auto Selects automatic play

▷▷ Next Image button-Selects the next song from the play list

◁◁ Previous Image Button-Selects previous song from song list

◁)) Selects ?

Audio Screen Views

In addition to the normal screens described above there are other views derived from the main audio screen. These are described below.

Diagram Ten   Audio screen heirachy
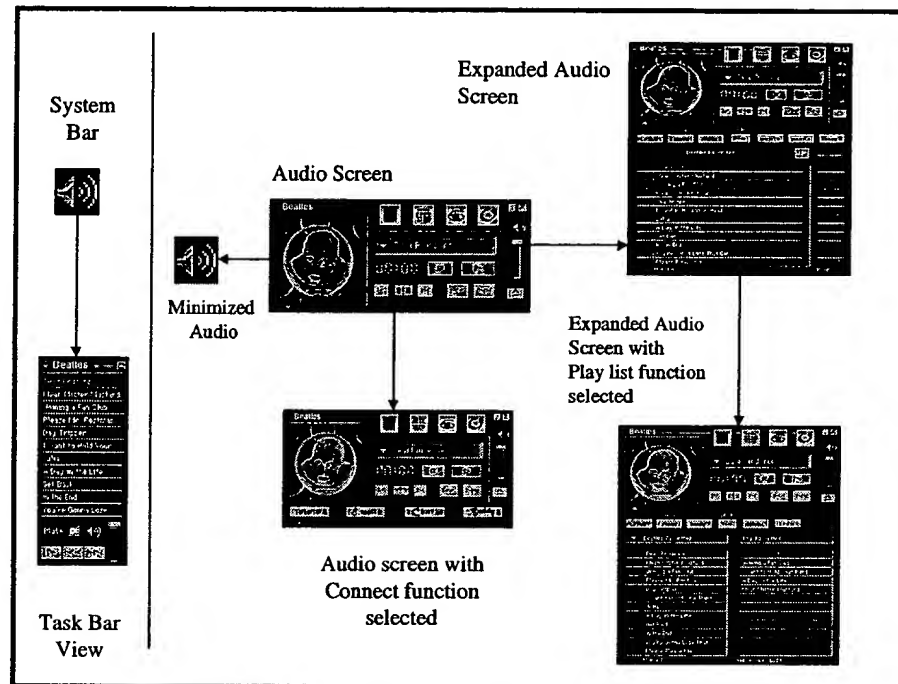
Expanded View

The expanded view will present the user with the functions required to not only play the audio but view the majority of information that is available to them

Diagram 11   Expanded Audio View
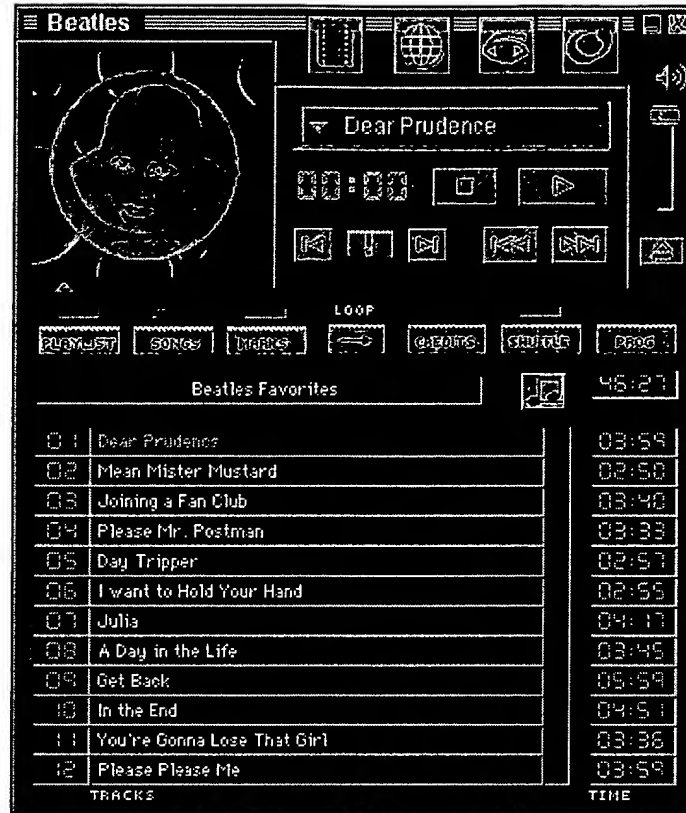
The expanded view will vary in size depending upon the requirements of the relevant functions e.g. A three song EP will only display those song titles, while a DigiBox with associated multimedia might expand to 800 by 600 on a PC platform

The main operational controls in addition to those described previously in the Audio player are:

Credits- Pop up menu of credits for song(s). This includes all the standard fields that may be searched by the find function.

Shuffle – This is a random play

Program – This opens the program screen to allow users to create their own play lists

Loop Button- this activates the loop function between ?

Accesses the sheet music function

The following buttons all have an indicator light above them that shows which button has been pushed and which information is displayed in the list window directly below.

Play list selector- These select and display the available play lists.

Songs Displays the available songs

[MARKS]   Marks- Displays the available locate points

Program Screen

The program screen provides access for the user to the Play list functions. This provides a mechanism for the user to construct their own play lists based on the song titles. These lists can be stored and replayed and may contain songs from a number of Digboxes. The diagram below shows the play list from the Audio screen, which is called the program screen.
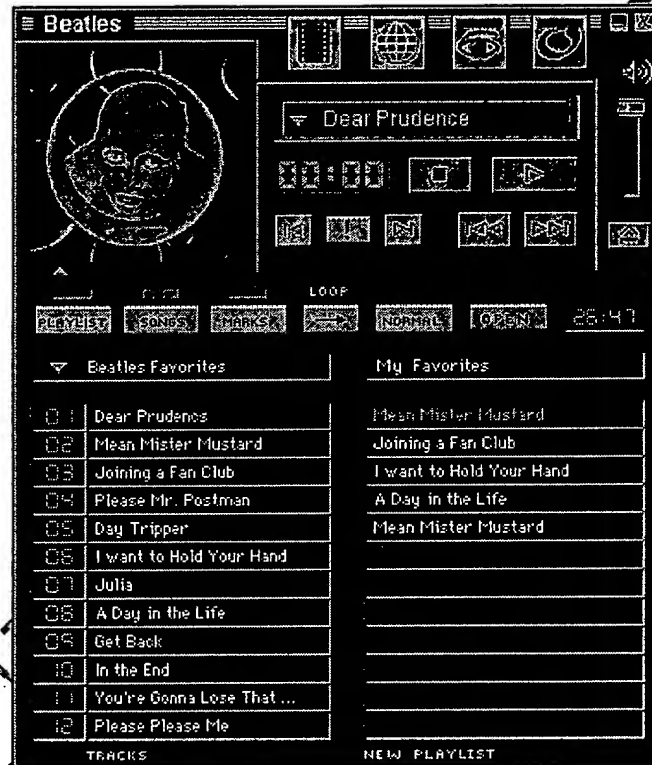
The operational functions provided in addition to the expanded audio screen are:

[OPEN]   Open Button provides access to the open screen

[NORMAL]   Normal button returns to the Extended Audio screen

Open Screen

This is the interface that provides access to the full scope and power of the player. The open screen will display information in the context from which it was called. The status of the Open screen will be displayed by the iconised meta tab being highlighted. All actions of the open screen will be context sensitive.

The open screen will provide the access to the EDL and Association functions and is the gateway to the players understanding of its own environment.

The open screen will display all the information sets of which it is aware and will provide the mechanism to display, search for and manage all the content of the user, even if that content is not held on their machine or owned by them.

The Open screen is effectively an interface to the data management system of the player and the digiboxes and their associated content.

Data management Structure

The data management structure is built on a database of all the data sets that the player is aware of, this includes;

Data Types	Description
DigiBoxes	the encrypted content store
	a database of a listing of their content (music, videos, graphics etc).
Associations	independent relationships between content
EDL's	Lists of events related to time and content

These data sets are accessed by the applications and the data provided to the relevant GUI's.
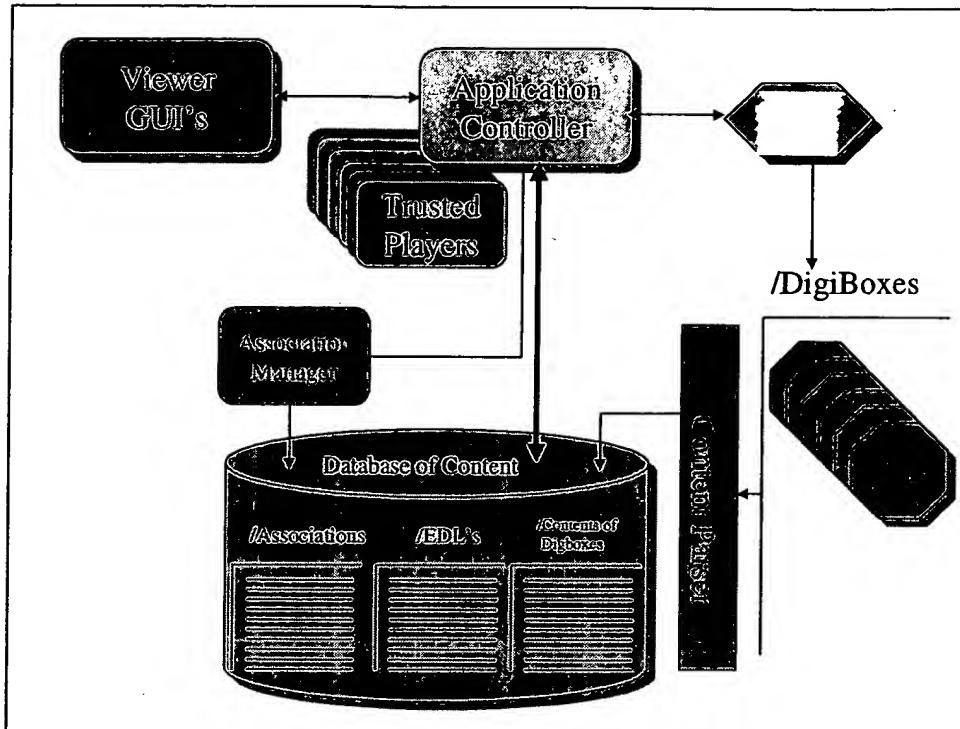
The structure of the data is dependant on the content contained within the DigBox .

Digibox Data Structure

Each Digibox will contain a number of data types or components.

Description	Types
Content	All content as specified
Business Rules	The rules for access to the content
Content Identifiers	The UID for each content component
Content Attributes	The Contextual information set for the content (i.e. Writer, Artist, Producer etc)

This data structure will utilize the format and descriptions of the ECD registration (Autorun.inf extensions). The initial data set will be:

- Artist
- Writer
- Producer
- Band Member
- Label

A custom field is also included.

The Open Screen

The open screen provides context sensitive functions to all the information known by the player. The Open screen is shown in the diagram below.
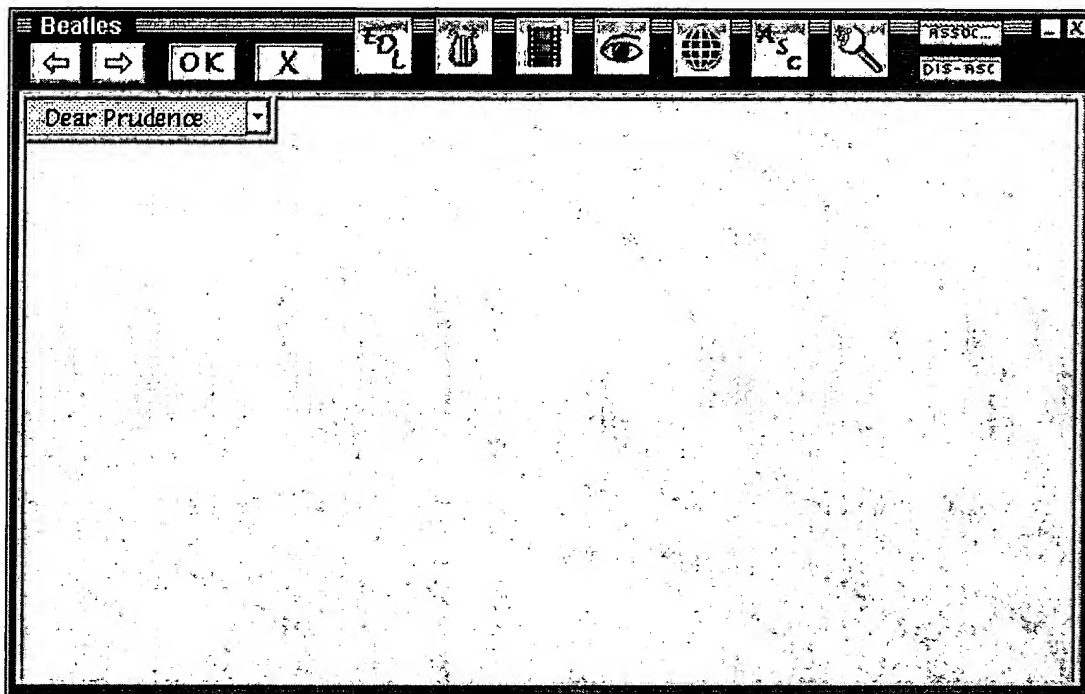
Open Screen Meta States

The types of Open states are identified by their Icons and are:

Music

Music will be accessed by the title of the play list. Each DigiBox will contain a clear text field that determines the play list of the contents. This will be in the format of the ECD play list file. This may be in the form of the album title, an "EP" or Single with say 3 songs or a single song. This will define the playing of the content of the DigiBoxes and may include Video or other media.

Videos

A listing of all videos known by the system Videos will be accessed with a list of those available.

Connect

Locations of music and content , headed by the default sets and including any user preferences. This will include both local and network connections.

Query(Find)

Provides users with a fast an efficient method to search and retrieve content, both locally and remotely. The find functions will be tailored to the requirements of music management on the net.

Associations
Providers and users can aggregate content. These aggregations and associations may be either local or remote.

EDL's
The method for accessing both Play list editing and location point editing and construction.

Each of these Meta buttons will have two information states, the default carried within the DigiBox complimented by that set of information which the user chooses to display. This information will be in the form of URL's and attendant icons

The operation controls of the Open screen are:

This

This

This agrees the selection(s)

This cancels the selection(s)

The Associate button spawns the association window.

The dis associate buttons disconnects the associations.

Some of these functions will spawn additional screens which are shown below:

Associations Screen

The Associations screen is spawned from the Open screen and provides a drag and drop interface between these screens. This will allow users to create and store the associations they require.

⇦　　　This

⇨　　　This

OK　　This agrees the selection(s)

X　　　This cancels the selection(s)
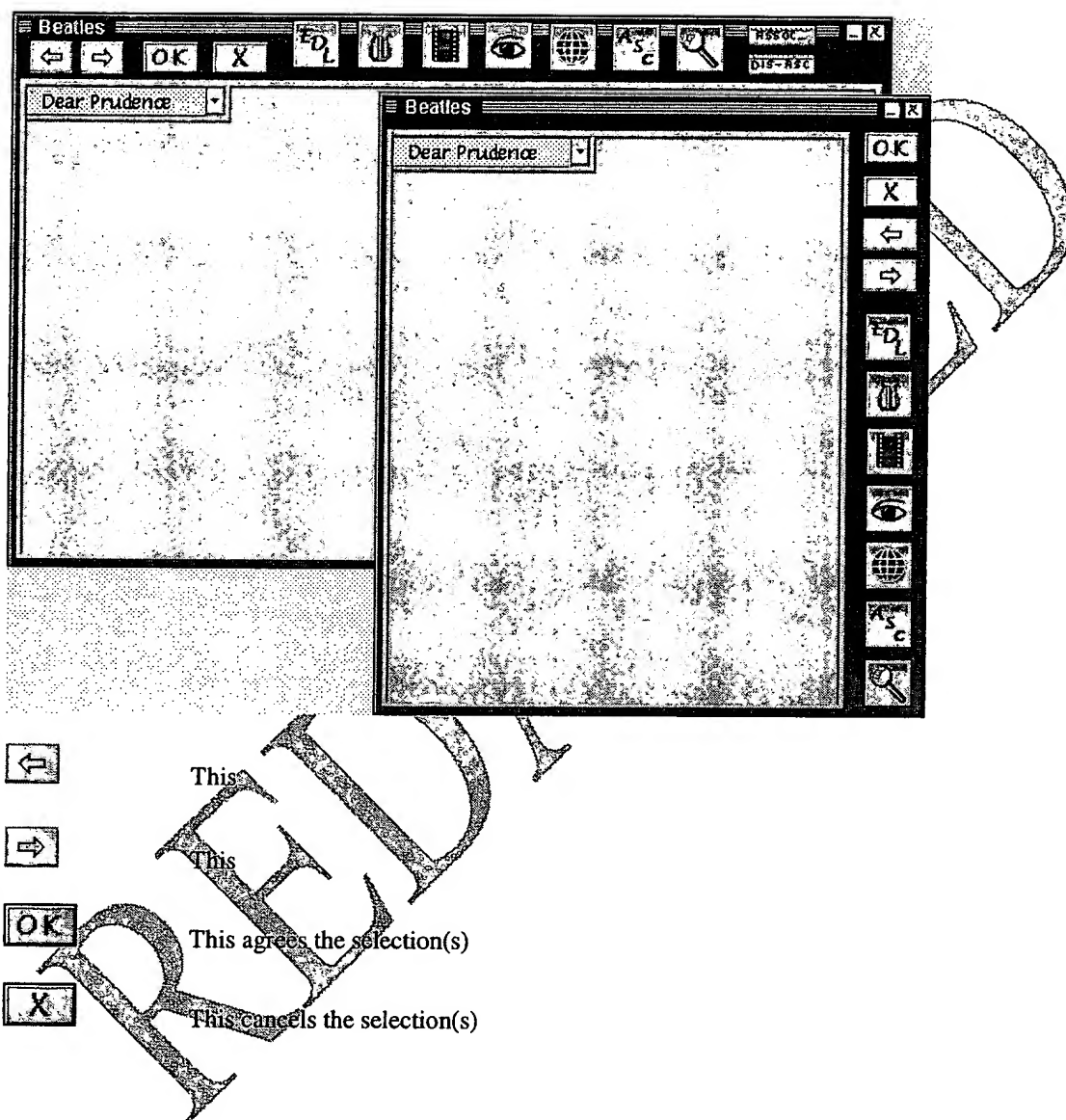
Find

This is the mechanism that the user is provided with for finding music content. The content is indexed via pre set fields, that are the output of parsing the content of the Digboxes. This will provide a fast and efficient method for retrieving information either locally or remotely. The Find screen is shown below.
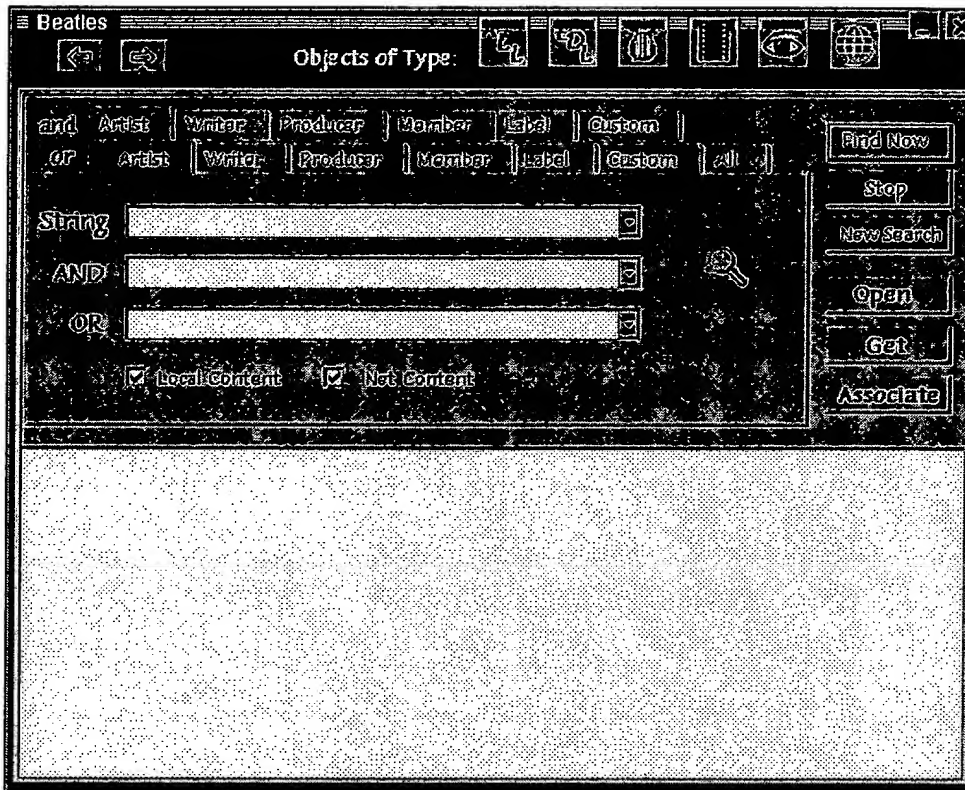
The find screen is designed to provide a rapid search for music either locally or on the network. The

The Find screen is able to find both locally and remotely, by interrogating the relevant databases.

The operational controls are:

Open This selects the open screen

Get This function selects the material for retrieval

Associate This creates an association for the items selected

⇦ This

⇨ This

Reference Icon's

Within the operation of the player a series of reference icons will be defined which will have the same meaning throughout the players operation. These are:

Retailers

Buying (More music?)

Radio

Preferences

The user preferences of the player will consist of two layers, the preferences for each screen and actions associated with each screen and the global preferences for the operation of all screens. The Global preferences will have precedence over the screen functions.

Access to the preference screens will be through the Open screen.

Help

All screens will have context sensitive help. Each screen will be able to access the standard comprehensive help functions supported by the Windows environment.

Application Components

[REDACTED]

[REDACTED]

[REDACTED]

Viewer Support Layer

To support the operations of the viewer and it's associated GUI's there is a support layer which enables it to interact with the e-commerce & security functionality provided by [REDACTED] and the and the supporting technology provided by [REDACTED]. The basic underlying functionality provided by [REDACTED] guarantees that all transactions related to the protected content in a DigiBox will pass through the [REDACTED] which will seamlessly communicate with the underlying framework over a network connection. A protected database will store information regarding the user's rights to specific content. The application will interact both directly with the [REDACTED] through [REDACTED] and the [REDACTED].

[REDACTED] provide the means for deployment of the [REDACTED] node on the user's machine, the users account management including payment history & budget management, and a backup and restore facility of the user's transaction records.

[REDACTED]

[REDACTED]

IMAGE REDACTED

[REDACTED]

Account Management & Payment History

The account management & payment history module is part of the ████████ modules and will allow users to verify and peruse their collection of rights to access content as well as the list of associated payments which have been made for the purchase of rights. This database of information will be continuously updated with data coming over the network originating from the ███ clearing house. This information will have to exist in a protected format; the database will be contained inside its own protected DigiBox and the records will be sorted by transaction date. Each record will contain:

- Transaction date & time
- Transaction price
- Content rights—this will be some combination of "own", "rent" or "trial"
- Rights expiration date if relevant, or 0 if not.
- Content Author
- Unique identifier of the content item; utilizing the underlying ████████ framework a unique alphanumeric identifier will be created for each piece of content during the packaging process.
- ISRC code if relevant
- Content name—in most cases a song, thought it could be anything
- Content compilation name—in most cases an album or CD name

For a given compilation of works the user who has rights to the compilation will have detailed individual rights to each content-component of the compilation work.

Users will be able to access their payment history, and contact the Master Service Provider (MSP) via e-mail directly from the application and/or over the phone if there are any problems. The UMP viewer application will forward all of the necessary details regarding the user's account to the MSP to facilitate any payment tracking issues.

Using a dispute mechanism which will be available in ████████████████, users will be able to initialize a dispute over incorrect charges or missing or incomplete rights.

Budget management (BM)

The Budget Management module is an ████████████████████ set of tools for the player. This object will provide information as to the availability of budget with which purchases may be made. It is anticipated the minimum information set will be:

- Total Budget available
- Spend to date
- Available to spend

The UMG component will set a level at 80% of the amount of total level of the budget and display this as a warning to the user that their budget allocation is about to be use up. This will then provide a function to activate the appropriate action to increase the budget.

Back up and Restore (BR)

The database of historical transaction records will be held locally in the secure database. This set of records may get corrupted requiring a back-up & restore methodology for the user. The back up and Restore functions may also include the storage of pointers to the relevant DigBoxes which are held on network servers for the user. The database of transaction records may grow to a large size in which case the user may not want it to occupy as

much space on their hard-drive—users will be able to back up old data and reduce the size of their historical database; they will also be able to restore any records—which may be needed for settlement of disputes.

Deployment

Controller

The player controller will coordinate the interaction with the content stored in DigiBoxes through the ███. The content will be accessed through a protected memory space that matches the certification guidelines provided by ████████. The content types include audio in PAC and DTS formats, Video in MPEG and QuickTime & AVI formats, graphic images in JPEG, GIF & BMP formats, text either as RTF or HTML or as images, as well as any executable and object files. The player controller will provide Component Object Model (COM) support so that it can perform simultaneous operations on different pieces of content. The time required to unpack content, and the predictability of the playback of the content(s) will be dependent on the performance of the ██ components.

The Controller is comprised of an Event Scheduler which receives messages from the Viewer GUI or from the EDL Editor to interact with content, which typically is protected and resides in DigiBoxes. Different types of content require different types of players. The Controller will run in a protected memory space and will interact with ████████ ActiveX Player Components. The Scheduler will attempt to synchronize timed events. These synchronized events are only available through the EDL functions.

The Controller will be able to synchronize the simultaneous play or display of content—the Controller will launch individual content decoders and will create (and destroy) threads which will monitor and correct the execution of Edit Decision Lists. In the first version of the UMP Viewer, timing will be dealt on a best effort basis--EDLs, which are the only mechanisms for simultaneous content display will be based on a MIDI file which will provide the basic synchronization means. The ability to synchronize content that is getting ahead of its schedule by slowing down its display will be provided—this functionality will not be relevant until the ████████ is able to give an accurate prediction of how long it will take to unpack any given DigiBox, or an item within a DigiBox. Currently, ████████ will provide an ████ Time which should parallel the machine's internal clock.

Connected through

(more DigiBoxes)

Secure Players
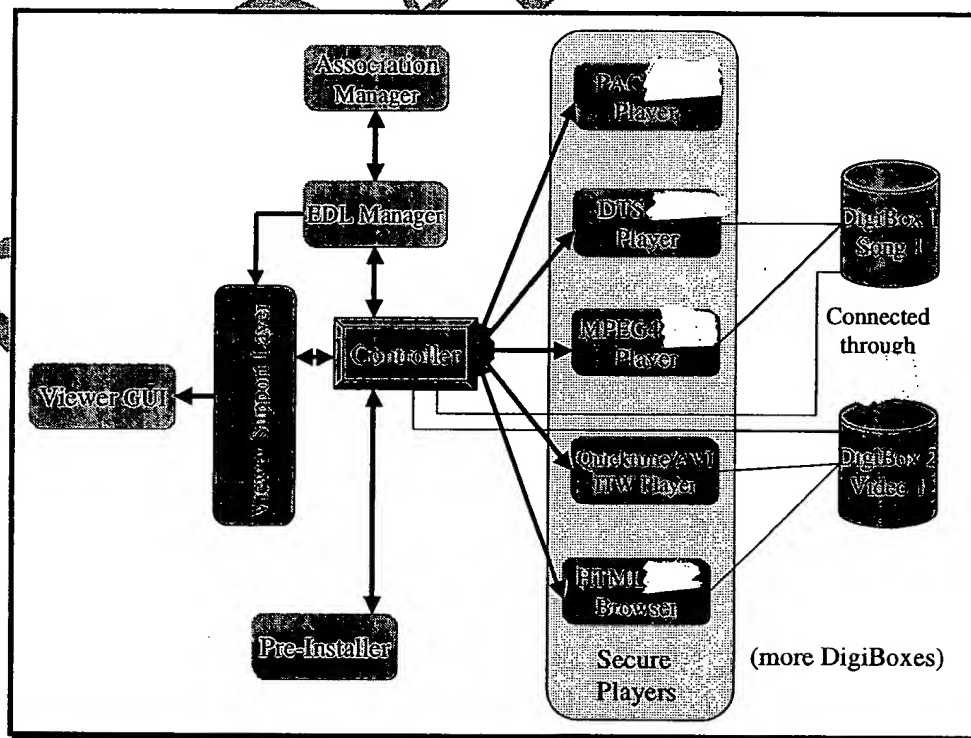
Dependencies

Decoders for all the different media types--audio, video, pictures & text--must be able to be fast enough so that they can work either sequentially or simultaneously without affecting the quality of the sound and yet being ████████████. These decoders, as stated in this document, are/will be developed by ████████████ or a third party—as is the case for the Sheet Music Viewer. The capability of the UMP Viewer to successfully play multi-channel Edit Decision Lists depends on the latencies & multi-threading capabilities of all of the individual decoders.

Success, for the initial version of the UMP Viewer, will be defined as the ability to simultaneously decode a PAC-encoded song, an MPEG3 encoded video and two JPEGS.

The viewer application will be designed to operate only under Windows '95 with PCs equipped with a Pentium or equivalent CPU running with a minimum clock rate of 100 Mhz. As much of the material will have a multi media content, the preference will be for the Intel MMX chipsets, which range from 133Mhz to 233+Mhz.

The UMP Viewer will support the following types of connectivity:
- 28.8KBbps & higher-bandwidth modems.
- Cable Modems, such as RoadRunner by Warner.
- PCs equipped with Network cards with direct connections to the Internet.

Browser

████████████████████ will be used to interact with protected HTML data. Non protected HTMLs will be able to be viewed using a standard browser.

The supported browsers will be IE 3.0, IE4.0, Netscape Classic and Netscape Communicator V4.04. In each instance if an older browser is detected, the user will be able to select a hot link to the relevant web site for an upgrade download. This will be undertaken in the installation script.

Chat & Buddy List

When users sign up for purchasing music with DigiBoxes they will be able to decide if they want to disclose their interests to affinity groups of other users utilizing the UMP viewer technology. They will be able to participate in dynamic chat communities and be able to communicate with users that allow themselves to be "seen" over the network. The UMP viewer application will store a protected set of chat group pointers, and collections of users. A 3rd party ████ application will enable the users to participate in chat, and will monitor other users who are connected and have given permission to be monitored. Users will be able to turn on or off (visible check list) their association with dynamic groups on the fly. Users will, dynamically, be able to make themselves available/unavailable to other fans listening to the same song or same EDL, and synchronize listening with them.

The chat server application & buddy-list monitor are a separate product from the UMP viewer application and need not be ████████████. Two preferred chat systems will be supported: E-Shares' Expressions[3] & AOL Instant Messenger[4].

The buddy list system should have the following key functionality:

---

[3] Refer to http://www.eshare.com
[4] Refer to http://www.newaol.com/aim

- Open architecture with multiple client interfaces; Active X, Java, Java Light, and HTML
- Easy access without any time-consuming downloads or plug-ins
- Powerful communications log
- Capture a written record of your online sessions
- Rooms: Create public, private, moderated, user-created rooms
- Buddy lists: Know instantly when your associates are online
- Overflow rooms: Can be created on the fly to handle large volumes of visitors
- IRC support: Supports the thousands of worldwide IRC users
- Follow: Users can automatically follow other users through across different rooms

Instant Messengers highlights are:
- Communicate instantly – It's fast, easy, fun, and free.
- Create your own Buddy List – locate your Internet friends online.
- Stay in touch -- exchange real time messages easily.
- Hold private conversations any time – day or night.

E-mail

██████ e-mail program developed by ██████ will be used as the main communication method with the Master Service Provider for account disputes and official notifications. For non-sensitive communications other e-mail programs will be supported—these programs will be Eudora, Microsoft Mail (Exchange) & Netscape Mail.

Users will be able to forward entire DigiBoxes as attachments to any other user. As is always the case, only users with rights to specific content will be able to interact with it. Portions of DigiBoxes will not be able to be sent to anyone—only entire DigiBoxes..

User created or copied Edit Decision Lists will be allowed to be sent to anyone and will exist in non-protected form. The EDL Configuration file will contain all of the information regarding how to purchase the rights to non-owned content so that the recipient of the EDL is able to enjoy the same experience as the creator of the EDL.

EDL System

The EDL system handles all the relationships between the content. These realtionships may have a temporal or non temporal natutre

Non temporal

The non temporal relationships are the associations which involve relating one orr more pieces of content to one or more other pieces of content.

Temporal

The temporal relationships have tow granularities, Play list and Locator points. Each of these has a temporal element. The play list is a sequence of events which happen in a specific order. The minimum granularity of a play list is a song. The Locator points are a sequence of events that occur within a song in a predetermined order. The minimum granularity of a locator point is defined by the timing method used. It is anticipated the timing method to be used will be MIDI.

Diagram XX The overall scope of the EDL system is shown below.
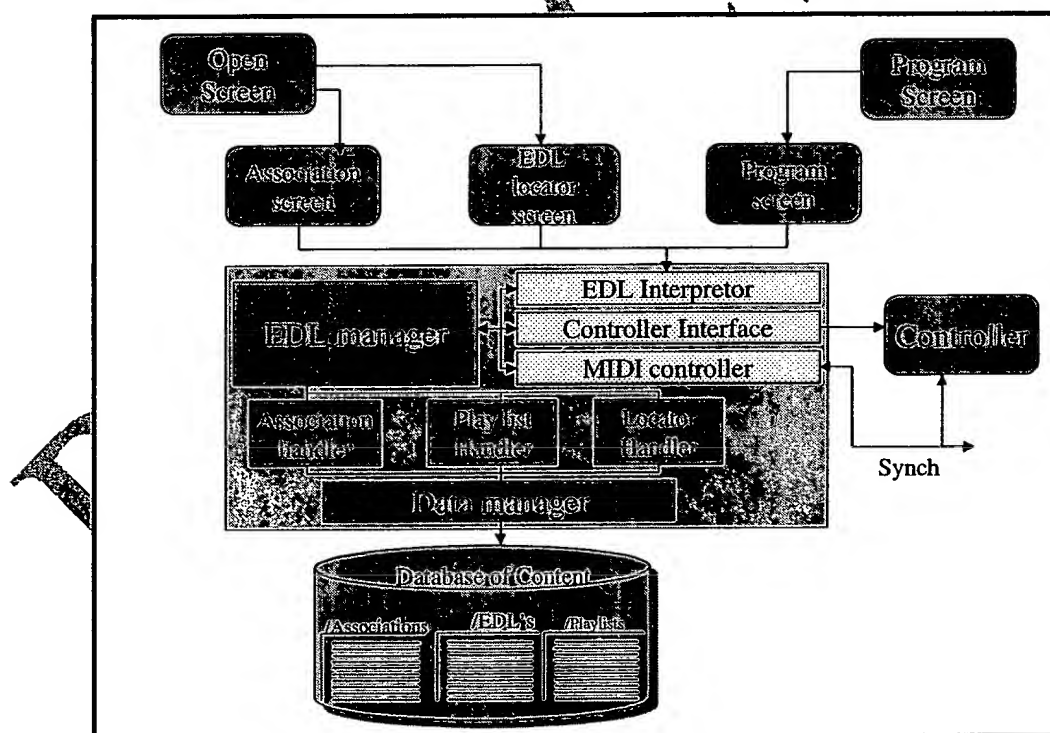
Play lists

These are accessed through the program screen in extended Audio mode. This offers the ability to construct lists of songs to be played in sequence and for these to then be stored and replayed as required.

EDL Locator points

Edit decision lists (EDLs) will allow the user to view a sequenced (sequential or simultaneous) presentation of any of the media types supported by UMP. EDLs imply the existence of a timing or synchronization mechanism. Initially the accuracy of synchronization does not have to be great. The MIDI file format is a good choice to drive the sequencing mechanism. The anticipated storage format may be flat (ASCII) files. The ███████ framework provides access to an Internal clock, referred to as █ time, but currently the time is not adjusted for unpack time—in other words, in the context of streaming data, the ███████ framework cannot inform a priori when a specific content item will become available.

There will be a graphical user interface that will allow the composition and testing of EDLs. EDLs will only consist of references to media contents, and not embed the contents themselves, therefore they will be relatively manageable and short in length. The EDL sequencing engine will be embedded in the UMP, and when users *open* EDLs they will be automatically executed by the sequencing engine.

EDL GUI

A graphical tool to produce & edit and playback Edit Decision Lists will be provided. This mechanism will allow users to compose multi-media arrangements of content which can be sent to other users. In order for the other users to view the composition they must have all of the necessary rights to the content.

An example of the EDL locator screen is shown below.
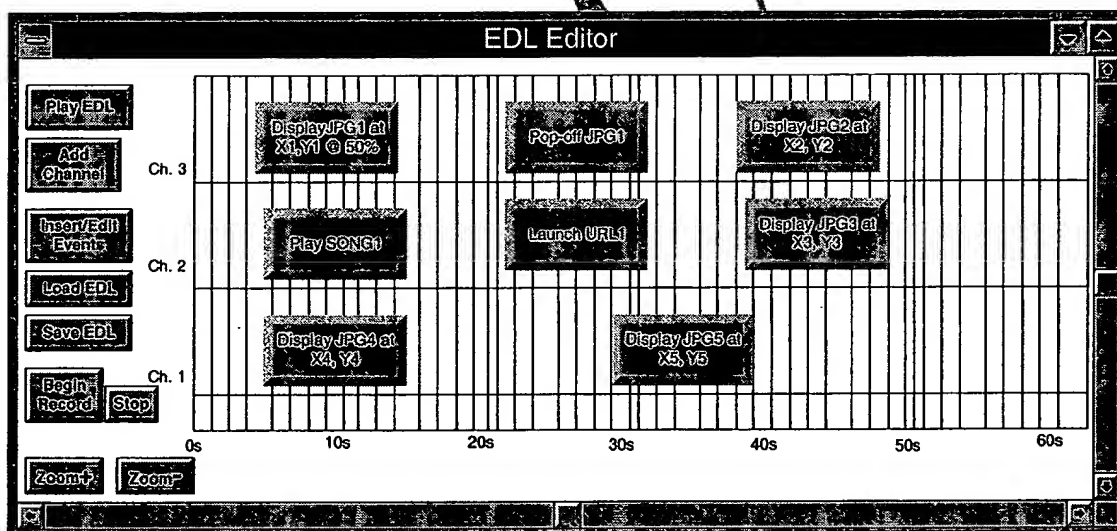
Associations Manager

Associations are non-synchronized relationships between content. They can be created by the user or supplied by the creator of the DigBoxes.. They can be simple associations of songs in no particular order, or they can be generic descriptions like: "All songs from the same author." The mechanism which finds all of the associated elements is referred to as The Associations Manager.

The Associations Interface allows users to input associations and thus aggregate content. The association will have two states, Local and Non Local.

Installer – Viewer Unpack

The UMP viewer application will be deployed by an install mechanism here referred to as Viewer Unpack. The most likely install scenario will consist of user downloading a single DigiBox. During the download process a simple install shield executable will be downloaded along with the DigiBox. The install script will figure out if the ███████████████████████ exists on the user's machine; if not it will automatically download an ██ and install it.

The UMP Installer is comprised of a Pre-Installer Module which determines, if any, which components need to be installed, and the actual UMP Installer which simply sets up the UMP Viewer executable on the viewer's machine.

Section Two: Functional Requirements

This page is left intentionally blank

Functional Requirements

This section deals with the functions supporting the architecture and GUI's.

Accessing any form of media with UMP will involve the "Open" function, which is tailored to the specific media type or collections of types. Each of the different views will have their own set of functionality; the screens are the Audio View, the Video View, the Connections View and Graphics View. The Customizations/Preferences/Transactions View should be handled as a dismiss-able, minimize-able overlapping spawned windows. . Additional functionality will be required by the Edit Decision List set of tools. Edit Decision Lists will be viewable in main windows but editing will require the spawning of customized windows.

Viewer Install

The UMP viewer application will be deployed by an install mechanism here referred to as Viewer Unpack. The most likely install scenario will consist of user downloading a single DigiBox. During the download process a simple install shield executable will be downloaded along with the DigiBox. The install script will figure out if the ███████████████████████ exists on the user's machine. If not it will automatically download an █ and install it.

The UMP Installer is comprised of a Pre-Installer Module which determines, if any, which components need to be installed, and the actual UMP Installer which simply sets up the UMP Viewer executable on the viewer's machine. When users download DigiBoxes they will be encapsulated with the UMP Pre-Installer as a self-extracting executable file, which will unzip the DigiBox & the Pre-Installer and execute the Pre-Installer. The following diagram specifies the details in the process:

Self-extracting file:
"dbx.exe"

Digibox

UMP Pre-Installer

UMP Pre-Installer Process

*Checks to see if UMP Viewer is installed.	If not inst. →	Downloads UMP Installer & runs it.
*Checks to see if InterTrust & SNS framework is installed	If not inst. →	Downloads SNS & IT installer & runs it.
*Checks Autoplay preference in the UMP Viewer.	If Autoplay is set →	Plays default EDL in the downloaded Digibox.

Open Function

The Open function is the gateway to the extended functionality of the player. The Open function has a series of Meta states indicated by the icons on the top of the screen. Every action of the Open function is context sensitive.

To simplify users interaction with the UMP viewer *all* related assets will be shown only when they can all fit in the "Open" window (unless over-ridden by user click) and the window will be sized appropriately to avoid both scrolling and white space.

DESCRIPTION of each sate, including its default and options

Data management

Content items can be associated in many different ways. Depending on the context in which the UMP viewer is in different choices will be made available to the user when he or she utilizes the open function.

Open choices will depend on whether the user has previously opened content DigiBoxes or not. If the user has not opened anything the open-dialog will be similar to this:

When the user leaves the mouse pointer over one of the open-choices--Digibox. Associations, EDLs, saved Messages, Query & "Get More Music"—for more than 1 second a popup menu list of all the possible values will appear. A maximum of 5 values will be displayed at once—the last cell in the menu-list will contain "...". If the user sets the cursor over the "..." for more than 1 second, then the values will commence to scroll upwards.

If the user has already opened a DigiBox, the open-choices will be augmented by Songs, Videos, Connections and Multimedia:

If the user selects the "Query" choice a query-box and a go-button will appear. Upon hitting go the query-box and the go-button will disappear and a window of results will appear:

DigiBoxes will be able to be opened by the name of the DigiBox. If the user has Autoplay on, then the Initiator module of the Controller will load a default Edit Decision List embedded in the DigiBox and will begin playing it.

Users may also be able to do queries to open content. Queries will access content contained in the user's machine or may also access databases which are linked over URLs which music distributors can embed into DigiBoxes. Users will also be able to explicitly search

Music

Songs and groupings of songs, i.e. an album , will be shown as a multi-state icon..

A single audio Icon, two to four times as wide as a single icon, that could be selected in different areas to highlight the: first and second, first second and third, or all four quadrants. Analogous to:

o)))

The three sizes would represent different granularity:

The three levels of granularity are:

Play Lists Lists of Songs
Songs
EDL Locator Points

Play Lists — Users can have any number of Play lists. The user will be able to have numerous play lists, each personalised for their application, e.g. one may have one song list to get going in the morning and another song list to for a mellow drive through the woods.

The play lists may be delivered within the DigBox having been created by the content provider or artist, created by the user, or appended by the user to other lists created by either the content provider, user, or another user or aggregator.

Play lists may become a property in their own right, such as radio stations focus their material on a specific demographic.

Songs   These are the basic core units of player, which may be broken down into loaction poits or aggregated into play lists

EDL Locator Points   These are the loactions within a song where synchronised events may be attached. The granulairty will be limited by the ability of the ▮ system to ptocvide timeing information. These will initially be driven as a MIDI program, with the user being able to define the loaction points. Creators may wish to provide pre determined points (e.g. Verse 1 , chorus2 etc) with the DigiBoxes.) The ability to resolve these pioints and undertake the associated actions will be controlled by the event controller.

Video
Connections
Graphics
EDL
Associations

for non-owned content, via the "Get More Music" Functionality.

Associations

By default all contents of a DigiBox will be considered related, under the name of the DigiBox, which will be its pre programmed playlist (or EDL).. A typical DigiBox may consist of a music album & some related artwork, as well as some URLs—in this case the name of the Digibox will probably be the name of the music album.

A protected database of associations will be kept by the viewer application on the user's machine—the associations can be generated by the user or supplied in the Digibox.The user will be able to append associations to those delivered within the the Digibox though the name will have to be different. For example a user may generate play-lists for which he or she has purchased rights to By default there will be a number of associations made:

- All songs.
- All videos.
- All URLs.
- All Executables.
- All songs by the same author/performer.
- All videos related to the same author/performer.
- All URLs related to the same author/performer.
- All sheet music.
- All sheet music for the same writer.

The user will also be able to create a special list of associations referred to as Non-Owned content (NOC) associations which contained individualized URLs for the purchase of specific content items.

If nothing is open in the UMP Viewer app Users will be able to open individual DigiBoxes by name or will be able to open the user's associations, including the previously described default associations.

If the user has already opened a Digibox, then the open function provides context sensitive choices in a window – the "Open" window:

- If the user is listening to a song that has a video attached to it will be displayed to the user as a possible choice to open.
- If there are no related or associated content items with the currently opened song other songs are displayed.

Icons will be used (across the bottom of the window) to change the object type, which will also be "real estate"/context sensitive, that is:

- If there is one video object associated with the current song and there is also one URL they would both be shown and the "related objects" icon would be highlighted (not grayed out).

- But if there are 20 songs in the album and 20 URLs only a couple of URLs and songs will be visible, and it would be shown that there are more songs and videos by showing a "..." icon and the "related objects" icon would be highlighted.
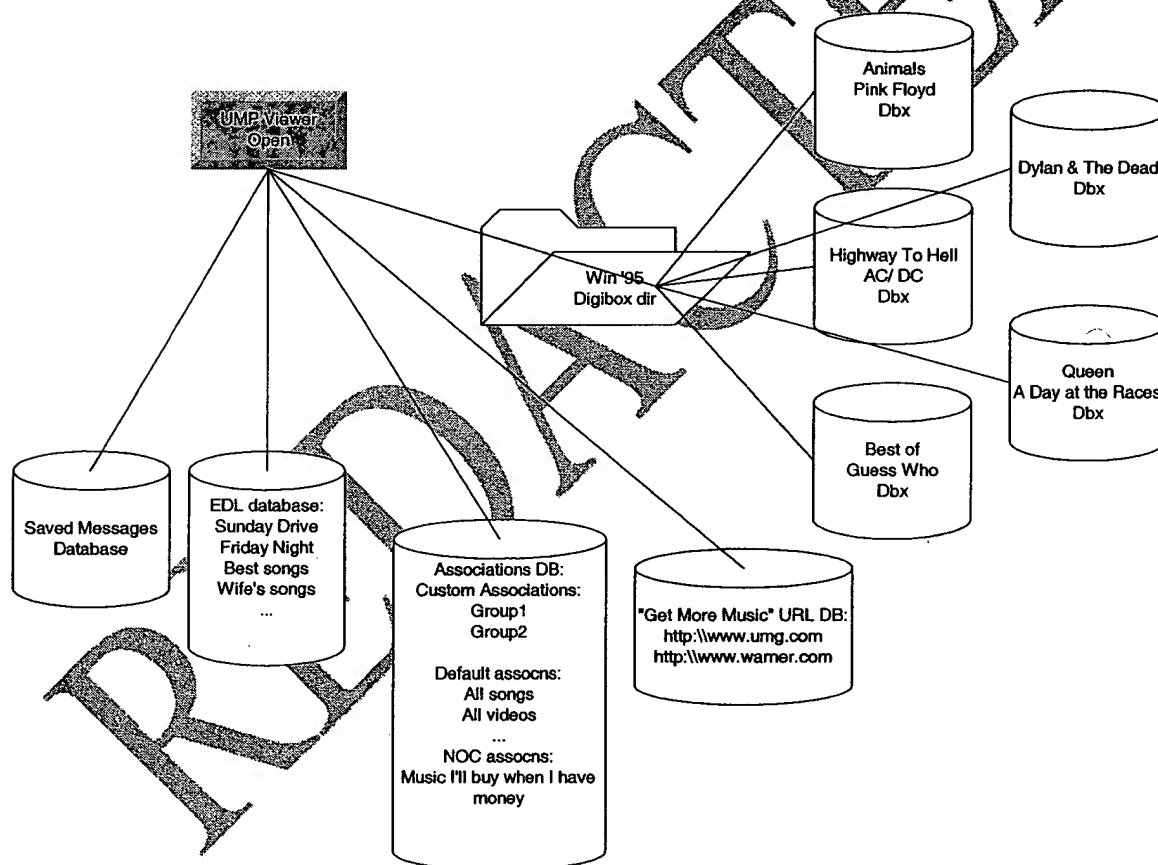
- **EDLs** (Edit Decision Lists)
EDLs have two basic levels of granularity:
    - EDL Play List
    - EDL Locator Points

EDL Play list manager.

Description and graphics of operation here

EDL Locator Points

EDLs can contain references to any data type supported by the UMP Viewer—music, pictures, video, text, URLs, executable object-files. Refer to the EDL section for more details.

- **Associations—Related Objects** for this song —
  - Pictures
  - Video
  - URLs
  - EDLs
  - Multimedia
  - Emails
  - Other songs in this album, by this artist, in this playlist
  - Audio Clips (guitar licks, sound effects, etc.)
  - Associations can be of local objects or non-owned remote ones. Associations will keep track of the rights to content—whether they are owned or not. Not-owned content will be indicated visually by "graying out."
- **Video**
- **Connections**
  - URLs
  - Email Addresses
- **Multimedia**

ExecutablesMessages (saved email, sent or received)

- **Get More Music**—Users will be able to access URLs embedded in DigiBoxes to sample music to be purchased.

Find Function

A third party search mechanism with a Software Development Kit which will support searching through flat text & text linked via URLs will be utilized to develop the query functionality. Special care must be taken so that the security of existing data in DigiBoxes is not sacrificed by this query functionality and that the search mechanism follows all of the necessary development guidelines to meet certification ▮▮▮▮▮▮—this implies that data results are not cached on the disk somewhere and are only contained in protected space.

Similar to an Internet search engine, users are able to search all of their contents. The results of the search will be displayed in a pop-up screen; if a user clicks on one of the results it will force UMP to open the selected content item. Besides supporting keyword searches, complex searches with the reserved tokens 'AND', 'OR' and 'NOT' will be allowed.

Find Get function

Icon   Get

This function will retrieve the identified remote object(s) and load them into the appropriate local store.

Clicking the icon will show the icon in a "depressed" state, as there may be a delay in retreiving the selected information.

"Get More Music" Function

Discuss this with Albhy

Objective of function

DigiBoxes will contain URLs that allow the user to purchase or sample more music. Each URL will have a text string which will name and/or describe the URL and a category string which the user can override and is mostly for sorting purposes. As the user downloads more DigiBoxes this list of URLs will grow as in stored in a protected database referred to as the "Get More Music" URL database. Users will be able to peruse or search for music content contained in these URLs. Users will also be able to produce association of non-owned music related content to be purchased at a later date. These associations will have embedded individualized links to
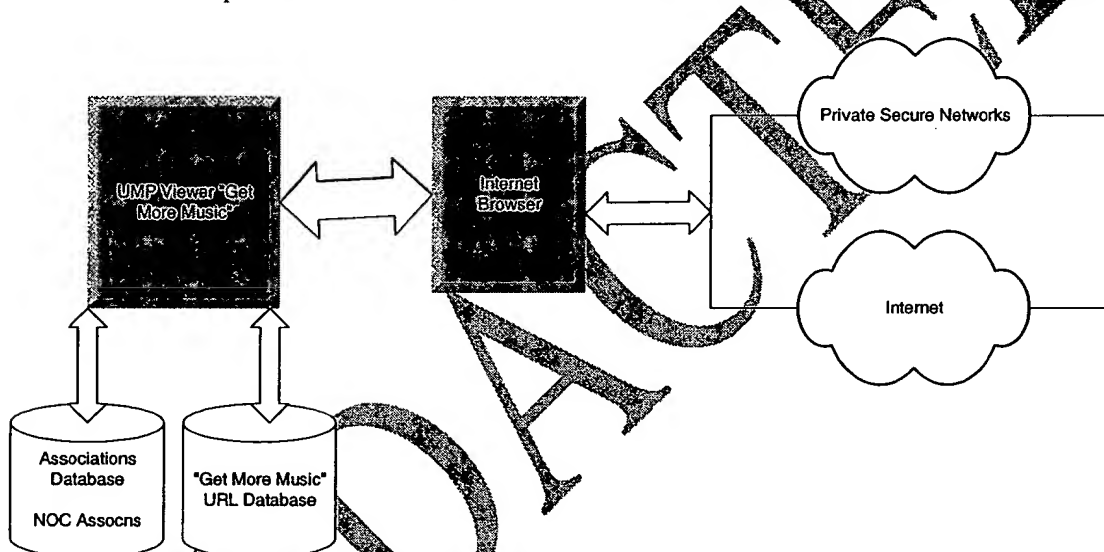
the specific content items that the user is interested in,
Getting more music (DigiBoxes) is facilitated in a number of ways.
- Find - can always be context shifted from local to connected
- Open - will have a new music choice which includes:
    - default URLs from the retailer, distributor of the currently loaded music
    - custom links of the users favorite acquisition sites
- Connections - will have getting more music as one of its default choices, launching the Open choices listed above

Associations Manager

Associations are the aggregation methods for the user to create user-defined relationships of content. Content items may be contained in many different DigiBoxes and may have a variety of ownership states.

Similarly to the way "Browse" buttons often work users will be able to:
- **Associate** to find assets that were either:
    - ⇒ Not in any DigiBox but that the user wanted to appear in the find windows or
    - ⇒ In a DigiBox that would not normally be associated with the current object (for example associating a Gavin (Bush) picture with a No Doubt song because Gwen and Gavin are an item).
- **Disassociate** to remove objects.

Path *—Right clicking on a name shows the path to a specific DigiBox if relevant.*

Meta Tabs *can perform open-like functions. If you have one video associated with a song, it might just jump to that when the video tab is pressed or it might spawn the open window. Right clicking will show a pop-up list of related assets of that content-type.*

Associations are a series of links between the content of one or more Digiboxes. Each piece of content of a DigiBox has a unique identifier (or reference number). This is assigned by the creator. Each piece of content may have associations to to one or many other content pieces. These references or associations may be established by the creator and packaged in the DigiBox.

Uploading into the Association manager

When a Digibox is downloaded, a process will interrogate the DigiBox to take every reference contained within the DigiBox and create the associations to other content held by the user. This will be done as the process references the UID of the content with the UID iod the users content.

The associations will be held in a datastore on the users machine.

The associations may have three states, Local, Remote or deferred. Both local and remote will be shown in the form of a URL indicating the loactions, whilst deferred will be shown as a greyed out refernece, which may be either a URL or another refernce or icon.

EXPLAIN DIGIBOX DATA

The data sets included in the Digibox are desribed below.

Universal Reference Point

This si the base reference point for the Digibox which will be expressed as a network location. This URL will then act as a server from which the user can be redirected to the appropriate resource.

This will be a text field containing a valid URL reference. The link for this reference will have to be available for a virtually unlimited time on a 24/7 basis.

Content UID

Each piece of content will have a Unique identifier. The form of this is yet to be defined, but is likely to be of sufficient length to provide for virtually unlimited content.

Each piece of content that is put into a digibox will have this UID. Th UID's may indiacte the source of the content (ie from a specific record company).

Digibox

Each Digibox has an identifying number. Each Digibox may be copied or duplicated and thus ther may well be many DB's with the same identifying number.

Distributro Field

Within the Digibox there will be a field to be filled in by the retailer.

Sheet Music Viewer

The viewing and printing of sheet music requires the use of a customized ████████████ viewer and printer engine. A third party with experience in sheet music publishing will do the development of this sheet-music application. Music can be viewed either as Sheet Music or Tablature. Viewing is one price (perhaps free for some content) and printing is a different price. The viewing engine should support synchronized viewing, that is; as the audio plays, notes or chords are highlighted. The playing of MIDI files is a separate application that may be supported in the future but will not be supported in the initial version of the player.

Screen Functionality

Each of the specific screens to be displayed to the user will support a functionality set. These are described below in terms of their proposed operation which will be developed as proof of concept designs and prototypes to refine and optimize their use and operation.

Since screen changes may change functionality without significant visible cues, a methodology will be adopted whereby new icons (or icons with different functionality) will appear highlighted for a few seconds after a mode change.

Audio Screen

Basic Functionality

Audio Functions
- Play/Pause
- Track Next, Previous, Locate Point Next, Previous — similarly to the "Open Songs" windows above (((O))), we can use an intuitive graphical granularity display for next locate point or next song (or eventually, next playlist).
- Scan Forward, Back or Location-drag-bar (Scan can function like many physical players where a prolonged click on a Previous/Next button makes it scan. This will be supported by visual clues.
- Volume
- Balance
- Eject (if physical media)
- Expand/Contract View
- Play list next/previous

Audio Display
- Artist Name (title bar)
- Song Title (w/arrow to display current song list) Rolling over the title displays the whole title if it is not fully visible

- Playlist or Album title
- Thumbnail
- Time (double or right-click drops window to change format, song elapsed, remaining, album or playlist elapsed, remaining)

Extended Functionality

Additional Audio Functions — *All of the above Plus:*

- Shuffle (Random)
- Repeat
- Create/Edit EDL (spawns window) — includes Program and Loop (multiple/storable). Lists can go from anywhere in any song to anywhere else (at millisecond resolution, in future versions).
- Auto-Locator — Display/Create and Label/Edit locations within a song
- Levels/Balance: — L/R, CD/WAV/Midi (launch Windows controls, we have our own level)
- Display/Print Sheet music — The first few lines of the sheet music are displayed. The whole thing can be purchased for display (synchronized with the audio) for one price or printed and displayed for another.

Video Screen

The functionality for the video screen is very similar to the Audio Screen's. When Video is   selected, the audio control should be modified as necessary. Video will be spawned in a new window or full screen. Users with the processing power to decode DTS should be able to display MPEG 1 videos as well.

Large-sized high-resolution videos, of at least as good quality as currently available DVD titles will be supported. The view-window may be as large as the entire user's screen.

Connections Screen

Basic Functionality

Connections' Functions/Display

— Replace some of the control/display space with three buttons for:

- URLs — go to Primary
- Email — Primary email
- Chat/Affinity(Buddy Lists) — extends the window or spawns a new window

Extended Functionality

Additional Connections' Functions/Display

— All of the above Plus:

- Subscription — centralized List Server for all our bands. We support three subscription levels.
1. Only let me know when a new record is released or they are playing in my town

2. Let me know about the most interesting things, TV shows, press releases, contests, etc.
3. Tell me every detail you have

There should be a simple list (on our server) with band names and radio buttons. On our (the Client) side, a Subscribe/Un-subscribe button takes you to the Subscription Web page at Universal, scrolled to the current artist. If you are not currently connected, a dialog lets you know that your membership will be updated when you next connect.

- URLs — extended, lists of relevant URLs. We should probably have a pop-up of a few relevant URLs and a few of the most recent. Our Open dialog shows all.

- Chat/Affinity Groups (Buddy Lists) — The purpose of Affinity Groups is to make yourself (the listener) available to other listeners who are listening to the same song/album/video. There is a radio/status button which sets the user to available/unavailable and a settings button that launches our Affinity software partners management window.

- Email — extended. Lists of relevant emails. Double-click or "Select and click send button" launch the users email client.

- The ability to **Attach Objects** to emails or chats. Things like:
   a) Locator positions
   b) EDLs
   c) Graphical elements
   d) URLs

Most of the above require having the DigiBox for playback. We might need an extension (mime type) for our object type(s). We have an object creation/collection tool and we drag and drop the objects into email or chat windows.

Media Screen (Graphics & Executables)

There are a few different sources for graphical material:
- *ECDs —Support the Architecture already defined in the extended Autorun.INF*
   ⇒ Play the Audio with as much of the DigiBox functionality as is relevant
   ⇒ Credits — read from Autorun.INF
   ⇒ Play the Video with as much of the DigiBox functionality as is relevant
   ⇒ Support URLs and Email in the same way they are supported now
   ⇒ Cover Art slide show
   ⇒ Multimedia and misc. object launch as per Autorun.INF
- *Digi-Boxes*
   ⇒ Cover Art slide show
   ⇒ Lyrics
   ⇒ Credits
   ⇒ Executable Multimedia (Shockwave, HTML)

Preferences/Customizations Screen

Users will be able to set and store configuration & operation details via this screen. They will be able to view and set preferences & settings, customize views and obtain help information.

One of the preferences will consist on an Autoplay feature which will play a default EDL which is embedded in a DigiBox automatically when opening that DigiBox.

██████████████████████████████ module which will inform the user which personal information the music providers have access to. This information will be accessed through the Preferences Screen. Users will also be able to choose from the preferences screen which kinds of personal information can be made public—to other users. This information includes age, real name, nick-name, etc.

Accounting Screen

Users will be able to check the status of their accounts for such things as credit levels, account history & balance, as well as being able to report any purchase or accounting problems. Most of the functionality that the user accesses with this screen lies in the Viewer Support Layer. Since the ████████████ is not responsible for maintaining the users' records, the viewer will maintain its own database. It is recommended that this database of records be stored in a DigiBox—this will not be a requirement in UMP V1.0 but it is suggested for future releases. Storing financial records in a DigiBox implies that UMP has the ability to package content with the necessary ████████ tool set required.

Edit Decision List Functionality[5]

The EDL file types will be set up as new MIME types in the installation process. There are tow instantions of the EDL functionality, the EDL play list and the EDL Locator points. Both are mangaers of events that have temporal elements.

Synchronisation

A key function of the operation of the EDL is the notion of synchronisation. This will be haevily dependant on the capabilities of the host players and the connection method between hosts. As the prime connection is likely to be the Internet, the strategy for achieving synchronisation of activities will have to be indirect.

The proposed method for establishing sync between tow remote players is as follows. The hosts will both reference the net time of their ISP's, or a common time server if available, to establish their internal PC clocks. The sync. Events will be run locally, requiring a previous download of the content and events to be played. The players will then begin at the pre determined time, giving a sync play, within the constraints of the timing mechanism.

EDL Playlists

Description and operation of Play list editor

EDL Locator Points

EDLs will be based on two files. The first file will be described as the MIDI-configuration file, and the second file will consist of a standard MIDI file.

---

[5] The same EDL functionality will be used by the Packager Application. See Appendix B.

The configuration file will specify, for a given number of MIDI channels, a number of media types to be played on each channel. For example a channel may consist of pictures, another one of videos, and a third of songs. The configuration file will also specify which MIDI metrics signify different aspects of the media-play—for example a note-on message might indicate to display a picture, and the specific picture will be indicated by the note-value. A note-off message will indicate that the last picture being displayed is turned off.

Since all content pieces will be protected they will have to be unpacked before they can be accessed. In this first version it will be assumed that the "performance" quality and timing characteristics that a user experiences during the composition & playback of the EDL will be roughly the same as another user who sees the EDL being replayed. The initial goal is to have a timing error no greater than 10ths of a second. In later versions this error will be reduced by employing a more intricate timing mechanism which will stall content items that are unpacked too quickly and will focus system resources on items that are behind schedule.
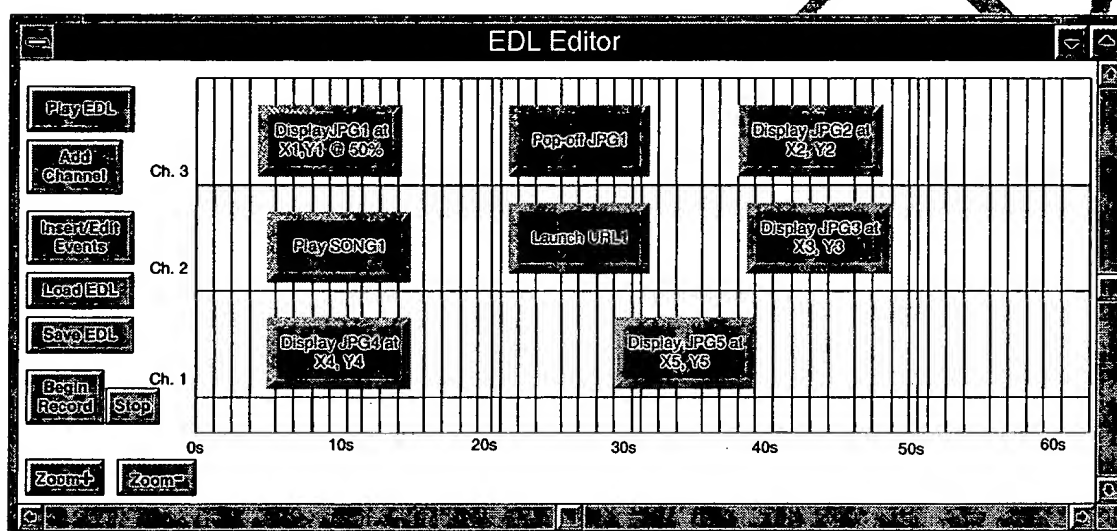
With the EDL Editor users will be able to compose & test EDLs. The basic functionality consists of:

- Playing the currently loaded EDL
- Loading an EDL
- Saving an EDL
- Adding channels—the number of channels is determined by the maximum number of simultaneous events
- Editing in Step entry mode
- Recording live from user's own interactions

Given that every media asset will have a unique ID that may be a very long alphanumeric string , before each multimedia sequence is "played" the media assets must be "mapped" to a numeric list of identifiers that can be easily coded by MIDI values—the mapping between MIDI codes and media assets need only reside in memory during the multimedia "performance" or these media assets may reside in multiple DigiBoxes.

Another reason that makes MIDI an interesting choice is that one of the supported types of media assets will be sheet music, which can also be represented in MIDI format—this functionality is reserved for a future version.

All windows machines will be able to have access to decent (software based) music synthesis.[6] MIDI files have very low bandwidth requirements and could be used, for example, as a teaser when downloading large DigiBoxes.

A user, with the aid of a simple click & drag based interface will be able to select content items upon which actions are performed in a chronological order. Users will be able to generate EDLs in two main modes:
- Step Entry
- Real Time Recording

Users will be able to create a location-point for specific places in a song. These can be used to step forward and back to favorite parts of a song. These locatin-points are ideal mechanism for dragging other events—such as the display of a picture—into an EDL.

With step entry users will be able to specify exact actions to occur at specific time intervals. The easier methodology constitutes of recording the exact users interactions with content and storing them for playback. Regardless of the input methodology, step-wise or real-time recording, the EDL will be able to be played back and further edited using both methodologies as well.

The specific parameters which the user can set are for different content types are discussed in more detail:

Music Parameters:
- [By default playing music will show an iconized play-controller.]
- Create Locator point (includes name function)
- Play from locator pointto
  - End of song
  - Next locator point (handling of loop functions- cross handling of loops from different songs)
- Play from mm1 minutes ss1 seconds tt1 tenths of a second to end of song.
- Play from mm1 minutes ss1 seconds tt1 tenths of a second to mm2 minutes ss2 seconds and tt2 tenths of a second.

---

[6] Roland and Microsoft have a partnership in 1997 where Roland has made available a software version of their SoundCanvas synthesizer free of charge to all Windows users, and will be included in further releases of Windows.

Section Three   Environment

REDACTED

This page is left intentionally blank

REDACTED